オブジェクト指向マルチメディアデータベースの永続性

オブジェクト指向マルチメディアデータベースにおける永続性

石　垣　久四郎

1. はじめに

マルチメディアデータベース技術は，文字列のみではなく，画像，図形，イメージ，音声など多彩な情報表現とその処理を対象とし，新しい記憶媒体技術を活用して，高速・大容量の構内ネットワーク（LAN）を介した分散システム環境の下に，データベースシステムを実現するための新しい技術である。これは，情報表現軸と新記憶媒体軸の高次的な結合が重要であり，多彩な情報表現の統一的な扱い方が研究開発の課題である。従来の画像データベースシステムでは，属性に基づいて画像の内容検索を行う方法であるため，画像中のオブジェクトについて様々な抽象化レベルに属性を手動で索引付けするには限界があった。新しい画像データベースとは，異種の画像データを同じ利用者インタフェースで共同利用が可能であること，同時に文字列データと画像データとを同じデータモデルによって表現し，且つ理解できることが重要である。すなわち，文字列データ，画像データ，音声データなどを統一的に表現でき，理解できるようなデータモデル（実世界の情報関連・構造を直裁にコンピュータシステムへ写像する）を具体化し，マルチメディアデータベース中のオブジェクトの意味を理解して検索を可能とすることである。

1990 年代当初から，オブジェクト指向のアプローチは，マルチメディアデータベースシステム技術として実現性が高いということから世界で多く

議論されてきている。そしてオブジェクト指向は，個々の要素から複雑なシステムを容易に構築できる技術となるべく提供された多彩な新しい情報処理技術として注目されている。近年，コンピュータシステムの利用形態の大きな流れは，ネットワーク環境下でのクライアント/サーバ・アーキテクチャ（client/server architecture）が，従来の汎用コンピュータより多く選択されるようになった。

これらクライアント/サーバ・システムの急速な普及の根幹は，マルチメディア・データベースサーバが情報ネットワーク上で並行に共有化可能な全情報のリポジトリ(repository)として，しだいに位置付けられつつある。オブジェクト指向データベースは，単に先進的なデータベース・アプリケーションであるだけではなく，コンピュータシステムによる共同作業全般の要求を満たすべく，知的なメディア処理環境を提供することにあると考える。それは，文字，数値，図形，画像，音声といったそれぞれが全く異質な(heterogeneous)データを同質化して統合するには，それらを全てオブジェクトとみなすことで可能となるのである。

データベース管理システムの機能は，永続的データベースに並列にアクセスし，更新できるようにすることが重要である。そのためには，データの長期間の永続性を保証するためにトランザクション，システム，メディア障害に対し，様々な障害回復方策を講じなければならない。ここでは，オブジェクト指向データベース技術の概要を説明し，オブジェクト指向技術の概念で重要な継承機能の永続性について概括・検討することにする。

2. オブジェクト指向データベースの概要[(1)-(7)]

オブジェクト指向データベースは，アプリケーション領域と意味的ギャップを取り除く，いわゆる実世界の情報関連をできるだけ密接にモデル化できるため，モデル中の実体間のリンクや関連(entity reletion-shipe)

が直接表現され，操作することが可能である。そしてオブジェクト指向データベースのモデル化能力は，抽象データ型付け(abstract data typing)，継承 (inheritance)，オブジェクト識別性 (object identity) といった，オブジェクト指向から得られたものである。ここで，抽象データ型付け(abstract data typing)，継承 (inheritance)，オブジェクト識別性(object identity)とは，以下のようなことを意味している。

① 抽象データ型付け (abstract data typing)；

抽象データ型付けにより，オブジェクトのインタフェースルーチンの実装を隠すことが可能である。また抽象データ型付けは，クラスという再利用可能な構成要素により複雑なソフトウェアシステムを構築することができるようにするものである。

② 継承 (inheritance)；

継承により，新しいクラスやソフトウェアモジュールの作成をすべて始めから設計する代わりに，より汎用性のある既存のクラス階層の上に構築することができるようにするものである。これは，特化(specialization)によりなされ，特化を相補うのが汎化(generalization)である。汎化により，既存モジュールの共通の構成要素が抽象化される。特化と汎化は，永続性ソフトウェアの拡張性，再利用性，そしてコード共有を拡張するものである。

③ オブジェクト識別性 (object identity)；

オブジェクト識別性は，アプリケーション中のオブジェクト群を，任意のグラフ構造を持つオブジェクト空間に組織化することを可能とするものである。これは，従来までのプログラミング言語とデータベース言語によってオブジェクトを参照するという手法と比較して，オブジェクト識別性の強力な記法が優れており，汎用性もある。

したがって，オブジェクト指向データベースは，プログラミング言語と

データベース管理システムとの間のインピーダンス不整合性を軽減することが可能である。一般に複雑なアプリケーションにおいては，データはデータベース管理システムからSQLなどのデータベース問い合わせ言語によって検索（取り出される）され，それからの操作（編集加工）は言語CやPL/1といった従来のプログラミング言語で記述されたルーチンによって行うことができる。つまり異なるプログラミング言語（CとSQLなど）におけるデータ型は，同じではなく相互に変換しなくてはならない。データベース言語によりデータにアクセスし，それから従来のプログラミング言語により操作する理由は，データベース言語が問い合わせだけに制限されているからである。多くの場合，データベース言語では複雑な計算は，ほとんどサポートしない。計算は，プログラミング言語によってなされている。すなわち，オブジェクト指向データベースは，通常オブジェクト指向データベース管理システムのデータ操作言語（data manipulation language）により，アプリケーションの一連の計算を実行するための必要な表現能力を提供することができるという意味で，より完全である。

データベース管理システムの進化と普及，さらにオブジェクト指向システムの展開，クライアント/サーバ・アーキテクチャの普及が，1990年代の時代精神を表す流れである。それは，ネットワーク環境下で情報を共有化する必要性と，メインフレームからダウンサイジングの流れがコンピュータアプリケーションの分野におけるデータベースサーバの登場を促進したといっても過言ではない。データベースサーバは，ダウンサイジングされたクライアント/サーバ・コンピューティングの根幹として位置づけられる。そうして，オブジェクト指向データベース管理システムは，データベースとオブジェクト指向技術を統合する新しい情報システムへの流れである。オブジェクト指向の言語やシステムのアプリケーションは，永続性や並行処理，トランザクションといったデータベース機能を，その情報環境

に要求している。これらの要求がオブジェクト指向データベースと呼ばれる強力なシステムを生み出したものであると言える。

2.1 オブジェクト指向とは

オブジェクト指向とは，個々の独立したサブシステムから複雑なシステムへ統合し，構成するための手法を容易にするためのソフトウェアのモデル化と開発の規範であると定義されている。直感的に言えば，オブジェクト指向は，実世界の情報関連をモデル化し表現するための概念や手段を提供するものである。そして，プログラミングやデータのモデル化においては，他の技法よりも利点が多い[3]。

つまり，通常のプログラミング技法では，実世界の問題を解くために生成されるプログラムは，最初は問題を記号化したものからなり，つぎにそれをコンピュータの言語に変換したものからなる。これに対し，オブジェクト指向の方法及び技法は，自動的にこれらの変換を処理することができる。その結果，問題を記号化したプログラムの大きさと比較すると，かなり最小となるという報告がある。したがって，エンドユーザ，アプリケーション開発者及びシステムプログラマに大きな利益をもたらすことになる。

2.2 オブジェクト指向データベースとは

オブジェクト指向とは何か，オブジェクト指向データベースとは何かに関しては，昨今合意が形成されつつあるが，まだ混乱していることも多い。しかし，オブジェクト指向データベースは，オブジェクト指向と，データベース機能を結合したものであり，オブジェクト指向は，実世界の問題を直載に表現し，モデル化できるものである。このことは，オブジェクト指向の構文によれば，モジュールの実装の詳細な隠蔽及び参照によるオブ

ジェクトの共有既存モジュールの特化により，自由自在なシステムの拡張が可能である。

データベースの機能は，アプリケーションにおける永続性の保証や，情報の並行な共有が必要である。つまり，データベースにより，オブジェクトの状態を永続させたり，プログラムの更新作業を行ったり，多数のユーザが並行に同一の情報を共有できるようになるのである。したがって，オブジェクトの利点と概念を，データベース機能に結合することであり，次のような枠組みで(特徴付け)，定義することができる。オブジェクト指向は，次のように定義される。

・オブジェクト指向＝抽象データ型付け
　　＋継承
　　＋オブジェクト識別性

データベース機能は，次のように定義される。

・データベース機能＝永続性（継承）
　　＋並行処理
　　＋トランザクション
　　＋障害対策
　　＋バージョン管理
　　＋整合性
　　＋セキュリティ
　　＋性能

そして，オブジェクト指向データベースは，次のように定義することができる。

・オブジェクト指向データベース＝オブジェクト指向
　　＋データベース機能

以上のことから，オブジェクト指向データベースは，オブジェクト指向

とデータベースという，二つの概念を拡張したものであり，その能力は，これら二つの技術を密接に結合した点にあるといわれている[4]-[6]。

関係モデルにもとずくデータベースシステムは，1980 年代に入って次第に一般化し，1990 年代に飛躍的に普及した。そして，本格的な高度情報化社会でのマルチメディア時代の現在，これに代わるデータベースデータモデルの機能は，意味データモデル(実体関連モデル)，複合オブジェクトモデル(抽象データ型付け)，に加えて継承とオブジェクト識別性が基盤となるデータベースシステムが希求されている。

3. 継承：永続性のレベル

オブジェクト指向データベースが操作するデータは，一時的または永続的のどちらであってもよい。一時的データは，プログラムまたはトランザクションの内側だけで有効であり，一度プログラムまたはトランザクションが終われば失われることになる。他方，永続的データは，プログラムの文脈の外に格納され，個々のプログラムの呼び出しより長生きする。

永続的データは，通常トランザクションによりアクセス，共有，更新されるデータベースからなる。永続性には，いくつかのレベルがあり，トランザクションの作業空間の範囲内で永続するが，しかしトランザクションが終わると無効になるオブジェクトもある。ユーザは，ログインを確立しセッション中に様々な環境変数(パス，表示オプション，ウィンドウなど)を設定する。トランザクションは，一般にセッションの中で実行される。もし，システムが複数プロセスをサポートするならば，同じユーザセッション中で，同時に複数のトランザクションが活動することもある。

これらのトランザクションは，すべてセッションオブジェクト(セッションの存続期間中だけ存続するオブジェクト）を共有することになるが，しかしユーザがセッションを終えるとき，セッションオブジェクトは無効と

なる。複数のトランザクションとセッションにわたって存続する唯一のタイプのオブジェクトは，永久的オブジェクトである。これは通常，複数のユーザによって共有される。したがって，これらのオブジェクト（データベース）は，トランザクション，システム破壊，メディア破壊さえ越えて存続し，データベースにおいては障害回復可能なオブジェトとなりえる永続性が存続することになる。

4. 永続性のための選択肢[8)-(10)]

一般に，どのオブジェクトを永続的に選択すべきかを設定するには，次の永続的エクストテント，到達可能性の永続性，永続的インスタンスという3つの方法がある。これらは，永続的オブジェクトを作り，識別するために使われる。

4.1 永続的エクステント

永続的エクステントの概念は，データベース管理システムでは常に基本的仮定である。従来のデータベース管理システムにおいて(特にリレーショナル)，ユーザがデータ定義言語(SQL の DML のような)を使ってスキーマを定義すると，定義は構造とエクステントを取り入れる。しかし，オブジェクト指向言語では，ユーザはクラス構築によってオブジェクトの構造を定義する。この時クラスがオブジェクトの構造だけではなく，オブジェクトの振る舞い(操作及びメソッド)を設定することを除き，クラスは PASCAL のように従来のプログラミング言語における型と似ている。つまりほとんどのオブジェクト指向データベースは，オブジェクトとそのメソッドの構造を定義するために，クラスを使うようになっている。データベース管理システムの本質的な機能は，ユーザがデータを格納し操作し，ある操作を反復できるようにすることである。したがって，クラスはすべて同じ

型であるオブジェクトのカテゴリーを表しているので，永続的クラスのエクステントを持ち，クラスを抽象的データ型のコンストラクタとして，且つクラスの全インスタンスの格納場所として扱うと有用である。

永続的エクステントをサポートするオブジェクト指向データベース管理システムは，通常クラスの永続的インスタンスをトラバース,ナビゲートするための反復子，あるいは他の構成要素を持っている。永続的エクステントをサポートするオブジェクト指向データベースは，オブジェクトを作るために以下の3つのメソッドを持っている。

① 永続的クラスのインスタンスが作られる度に，インスタンスは自動的ににエクステントに挿入される。

② オブジェクトは，作るときにも永続的にするよう指定できる。たとえば，オブジェクトのNEWコンストラクタで指定できる。

③ 永続的データベースにオブジェクトを保存するためには，陽な書き込みまたは保存操作が要求されることもある。

4.2 到達可能な永続性

到達可能な永続性の方法は，一つ以上の永続的データベース・ルートを持ち，これらのルートから到達可能なすべてのオブジェクトを永続的にすることである。プログラミング言語あるいはデータベース言語は，従来のリレーショナルモデルのタプル（レコードなど）やコレクションオブジェクト（集合，部分集合，グループなど）のための様々な型コンストラクタを取り入れている。したがって，オブジェクト指向データベースシステムにおいても，この集合・タプルモデルを使って推移的に定義できるようになっている。図1.にトランザクションでアクセス可能な一時的オブジェクト空間と永続的オブジェクト空間を示す。

永続的オブジェクトは，一時的オブジェクトの部分オブジェクトになり

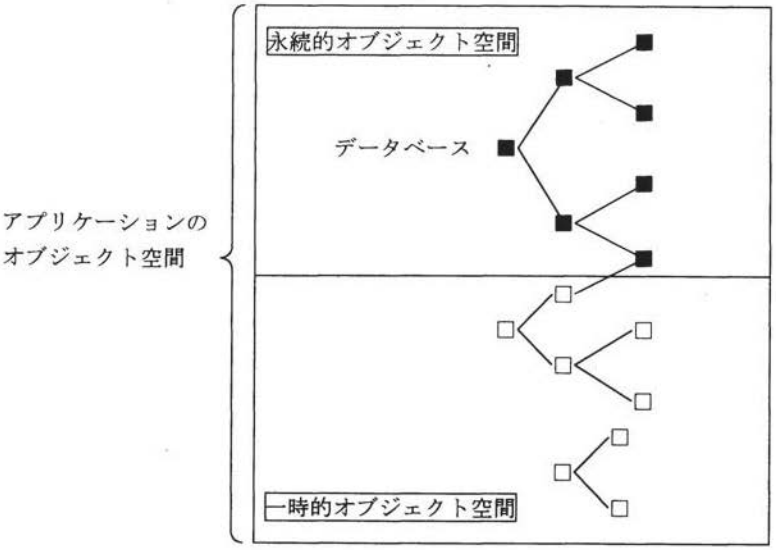

図1. 一時的オブジェクト空間と永続的オブジェクト空間

えるが，永続的オブジェクト空間の定義により，逆の現象が起こってはならないことに注意しなければならない（永続的オブジェクトのどの部分オブジェクトも永続的である)。つまり，永続的オブジェクトは複数の親を持つことができ，その内のいくつかは，一時的でもよい。そして一時的オブジェクト空間中のオブジェクトは，現在のトランザクションの範囲内でだけ可視である。したがって，トランザクションが終了すると，図1.の下側にある一時的オブジェクトは消滅することとなる。

4.3 永続的インスタンス

永続的インスタンスの方法は，陽に永続的であると宣言するか，あるいは既存のオブジェクトを関数呼び出しにより永続的にするかによって，クラスの特定インスタンスを永続的にする技法である。このアプローチでは，

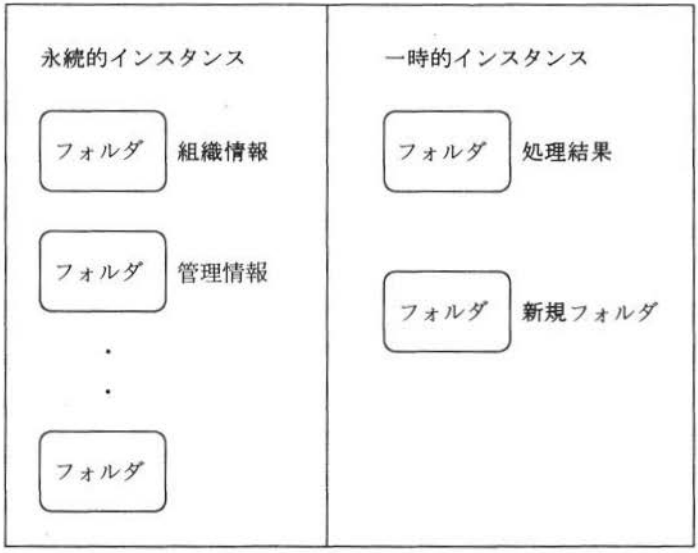

図2. Foderの永続的又は一時的インスタンス

永続性はクラスの属性でないことである。また，多数のユーザは，それぞれのデータベースを構築し，図2.のように各々のデータベースにオブジェクトを置くことができる。

同図において，この一組のフォルダは，クラスFolderの実行時に存在する全インスタンスの集合を表している。ユーザまたはアプリケーションプログラム開発者が，あるインスタンスを永続的と宣言し，他を永続的でないと宣言したものである。どのオブジェクトを永続的にするかについての指定は，クラス定義とは完全に独立であるということである。したがって，この機構を到達可能性による永続性のアプローチと一緒に使うと，正常に動作し，いくつかのエントリポイントまたはルートオブジェクトは，他のオブジェクトへの参照を持ち，あるいは他のオブジェクトの集合を含むことになる。また，これらのオブジェクトは，到達可能性によっても永続的

である。

4.4 永続的オブジェクト空間

永続的オブジェクト空間とは，永続的であるすべてのインスタンスのコレクションを指すということである。オブジェクト指向システムでは，インスタンスのコレクションはクラスのインスタンスであると定義し，クラスとその永続性との関連については，次の2つのアプローチが用意されている。

① 永続的インスタンスまたはオブジェクトは，永続的クラスのインスタンスでなければならない。つまり，永続的オブジェクトをインスタンス化できるクラスのカテゴリが存在し，永続的クラスは永続的であると陽に定義してもよいことになる。さらに永続的ルートクラスから直接的又は推移的に定義することによって永続的にすることができる。

② システムは，あるクラスを陽に永続的と指定しなくとも，永続性をサポートすることが可能である。つまり，どのクラスあるいはすべてのクラスが，永続的インスタンスをインスタンス化できることを意味しているもので，永続性はインスタンス又はオブジェクトに固有の属性であるということである。

5. 永続的オブジェクト空間（オブジェクト識別性）

オブジェクト指向技術とプログラミング言語の永続性における基本的側面は，参照又はオブジェクトポインタを永続的にできるということである。ps-Algol のような永続的言語は，ポインタ型を含むどんな型にも永続性をサポートすることができるようになっている。ポインタは，グラフ構造のオブジェクト空間を構築し，オブジェクトを参照することにより共有化す

るための有効的な手法である。

オブジェクト識別性は，参照によって共有化をサポートするための一貫した機構を提供するものであり，そして永続的オブジェクト・システムの具体化の重要な一つはオブジェクト識別性の実装技術である。オブジェクト識別性をサポートする技法の中に，識別子キーの使用，識別性としての仮想又は物理アドレスの使用がある。つまり，効果的なシステムを具体化するための技法は，この二つの手法を使うことである。

5.1 永続的オブジェクト空間と一時的オブジェクト空間

一般に，プログラミング言語は，永続性をもつように拡張されている。オブジェクト指向技術においても，永続的オブジェクト空間を備えておくべき基本的理由は，永続的データベースとプログラミング言語をサポートするためである。さらに，永続的又は2次記憶上の識別子をサポートするということは，より大きなオブジェクト空間へのアクセス機能を提供することにある。

5.2 アドレスと間接指示

アドレスに基づくスキーマと間接指定に基づくスキーマとのトレードオフは，オブジェクト移動の容易性や自由度とオブジェクトの構成要素にアクセスするためのオーバヘッドとの関係にある。対象とするアドレスには，次のようなものがある。

① 仮想記憶アドレス

② 2次記憶アドレス

③ 分散環境構造化アドレス

また，間接指定には次のようなものがある。

① メモリ常駐テーブルによるもの。

② 2次記憶常駐オブジェクトの索引によるもの。

(1) オブジェクトテーブルによる間接指示

各オブジェクト識別子は，オブジェクトテーブルのエントリへのポインタや索引である。初期の Smalltalk の実装[11][12] は，オブジェクト識別性を実装するためにオブジェクトテーブルを使った。

オブジェクトテーブルによる間接指定の方法は，余分なメモリアクセスと処理のオーバヘッドを伴う欠点があるが，しかしオブジェクトをメモリ中で自由に移動できるという利点がある。オブジェクトテーブルによる間接指定は，主としてメモリ常駐のオブジェクトに使われ，スキーマはオブジェクトを2次記憶との間で相互スワップするために仮想記憶システムと結合されることになる。

(2) アドレスによる識別性スキーマ

一般に，オブジェクトの識別性を簡易に実現するには，そのオブジェクトのアドレスを識別性として使うことである。これは，言語 C++ や Eiffel などのオブジェクト指向言語では，オブジェクトの参照機能に多く使われている。オブジェクトの識別性として最も有効的な手法は，永続性を妨害しない仮想アドレス法である。その理由は，以下のような2重の表現と永続的記憶が可能であるからということである。

① 2重の表現；

2次記憶上にある永続的オブジェクトのために，オブジェクトの表現を持ち，且つオブジェクトが主記憶に読み込まれるとき，それとは異なる表現を持つことができる。つまり，1つの識別性のスキーマが，ディスク上のオブジェクトのために使われ，オブジェクトが主記憶に読み込まれたとき，RAM 上の表現に変換されるということである。これを実行する最も簡易な方法は，オブジェクト状態の各変数をみて，各2次記憶上の識別性をその主記憶上の識別性に置き換えておき，必要があれば読み込むというもの

である。つまりオブジェクトが参照されたときのみ，変換を行うという方法が効果的である。

② 永続的記憶；

仮想アドレス法は，永続的記憶のアドレス空間を持ち，他の仮想アドレス空間と同様に使うことができる。永続的アドレス空間は格納されたオブジェクトを使用・参照・アクセスすることは，普通のメモリ上のアドレスと同じであるということである。また，アドレスとしての識別性としは，次のレコード識別子による識別性，構造化識別子による識別性，サロゲートによる識別性がある。

レコード識別子による識別性は，従来のリレーショナル型データベース管理システムにおける内部レコード識別子またはタプル識別子が，リレーションの個々のレコードを識別するために導入された手法である。

構造化識別子による識別性は，オペレーテングシステムにおけるパスの命名規則に似ており，特に分散ファイルシステムにより管理されているワークステーション環境下で有効的である。分散システムのファイルの識別子は，構造の一部がディスクやサーバのようなオブジェクトの位置的側面を把握できるように構造化されている。

最も識別性を強力にサポート可能な方法は，サロゲートによる識別性である[10]。サロゲートは，オブジェクトの状態やアドレスとは完全に独立したシステムが生成するグローバルな唯一の識別子である。そして，これは永続的な識別子であり，永続的記憶の中のオブジェクトにアクセスするために利用できる。

サロゲートによる識別性の具体化は，オブジェクト識別性の値・位置・型・名前から独立性を直接反映させ，各オブジェクトはどのような型でも，それがインスタンス化した瞬間グローバルに唯一のサロゲートに結合され，オブジェクトの存続期間全体にわたって，オブジェクトの識別性を内

部で表現するために利用されることになる。

6. おわりに

現在，永続的オブジェクト指向データベースは，その標準化が進められている。これまで概説したように，この手法を具体化するための永続的オブジェクトの定義と操作は，以下の機能の検討開発を推進することである。

- 従来のデータベース管理システムのデータモデルに代わって，新しいデータモデル/データ言語のアプローチを進める。
- 従来のデータベース言語にオブジェクト指向機能を拡張する。
- 既存のオブジェクト指向プログラミング言語にデータベース機能を拡張する。
- 拡張可能なオブジェクト指向データベース管理システム・ライブラリを提供できるようにする。
- 従来の親言語にオブジェクト指向データベース言語の構成要素を導入する。

現在，これらの課題に関連して，C++クラス階層，SQLのオブジェクト指向への拡張，知的メディア処理（内容に基づく検索）の実装化に目標を置いて研究開発が進められてる。この研究の適用範囲は，電子図書館，電子美術館，アートコレクション，建築・インテリア設計，地質調査，気象予測，医用画像などへの応用が期待されている。

参考文献

（1） 植村俊介；“新世代データベースシステムに向けて”，信学論（D-1），Vol 1. J74, No. 8, pp 443-446 (1991)

（2） Setrag Khoshflan；“OBJECT-ORIENTED DATABASES”, John Wiley & Sons, Inc. (1994)

（3） Atkinson, M., Bancilhon, F et al；“The Object-Oriented Databese

System Manifesto" In Bancilhon et al. (eds), Buiding an Object-Oriented Database System, (1992)

(4) Brown, A.W.; "Object-Oriented Databases." Applications in Software Engineering. New York: McGraw-Hill (1991)

(5) 出稿; "オブジェクト指向技術とマルチメディアデータベース", 東北大学附属図書館研究年報 29, 東北大学附属図書館, (1996)

(6) Cattell, R.G.G., "Object Data Management, Object-Oriented and Extended Database Systems", Reading, MA: Addison-Wesley (1991)

(7) K.C. Kim, "Performance of query optimization heuristics in object oriented databese systems," Proc. 2end Int'I. Symp. on Database Systems for Advanced Applications (DAS. FAA '91), PP 99-108, April (1991)

(8) Setrag Khoshflan; "Intelligent SQL." Computer Standards and Interfaces 12(1-3), Preseted at the X3/SPARC/DBSG OODB Task Group Workshop, Atlantic City, N.J, (1990)

(9) Setrag Khoshflan; "Modeling With Object-Oriented Databases." AI EXPERT Databases." AIEXPERT, October. (1991)

(10) Bancilhon. F., Briggs. T., Khoshafian. S, and Valgaeiez. P; "FAD-a Simple and Powerful Database Language", Proceedings of VLDB, (1987)

(11) Krasner. G.; "Smaltalk-80": Bits of History, Words of Advice. Reading, MA, Addison Wesley (1983)

(12) Goldberg. A. and Robson. D, "Smaltalk-80": The Languag and its Implementation. Reading, MA, Addison Wesley (1983)

(13) マルチメディア論文特集編集委員会"マルチメディア論文集"信学論 (D-II), Vol. J79-D-II, No. 4, (1996)

(14) 斉藤信男他訳; "プログラミングC++第2版", 凸版印刷, (1993)